

**Combining Qualitative and Quantitative
Spatial and Temporal Information
in a Hierarchical Structure:
Approximate Reasoning for Plan Execution Monitoring**

Louis J Hoebel

Rome Laboratory / C3CA
Griffiss AFB, NY 13441
hoebel@aivax.rl.af.mil

Dept. of Computer Science
University of Rochester
Rochester NY 14627-0226

Abstract

The problem of plan generation (PG) and the problem of plan execution monitoring (PEM), including updating, queries, and resource-bounded replanning, have different reasoning and representation requirements. PEM requires the integration of qualitative and quantitative information. PEM is the receiving of data about the world in which a plan or agent is executing. The problem is to quickly determine the relevance of the data, the consistency of the data with respect to the expected effects and if execution should continue. Only spatial and temporal aspects of the plan are addressed for relevance in this work. Current temporal reasoning system are deficient in computational aspects or expressiveness. This work presents a hybrid qualitative and quantitative system that is fully expressive in its assertion language while offering certain computational efficiencies. In order to proceed, methods incorporating approximate reasoning using hierarchies, notions of locality, constraint expansion and absolute parameters need be used and are shown to be useful for the anytime nature of PEM.

1 Introduction

The problem of plan generation (PG) and the problems associated with plan execution monitoring (PEM) have different temporal and also spatial characteristics in most cases. Only rarely is quantitative, metric or absolute temporal information available during PG. Most often temporal and spatial information is given as relation constraints between temporal objects and spatial objects. During plan execution the simplest and most intuitive form of information available will be metric temporal and spatial data. Some fact or event will be observed, reported or in some manner known. Monitoring the progress and continuing plausibility of a plan during its execution requires relating this metric, point information with the mostly symbolic, qualitative information that the execution is based on. A further consideration during PEM is that of bounded computational resources, which may not be present during PG.

Consider a complex system with a manager and multiple execution agent's. Further, let the agents be semi-autonomous (SA). They have some knowledge and

deliberative ability enabling limited action on their own. The problem is how to monitor an executing plan for the continuing basic physical plausibility of the plan's actions and effects.

We do PEM to determine the validity or plausibility of a plan successfully executing. For negative determinations, it's important to be able to control plan execution in terms of halting execution, selecting options, replanning or exhibiting some other desired behavior. This is problematic when the plans may be denoted with qualitative symbolic representations and the update information available during execution is mostly quantitative data points. Detailed micro-monitoring and constant translation between qualitative and quantitative representations may require more bandwidth and computational resources than are available. Coarse grained monitoring may miss subtle interactions and interconnections.

The need and use of temporal modification in plan reasoning is well established [McDermott82, Allen&Koomen83, Dean85]. Alternatives to state based paradigms have considered planning as qualitative temporal constraint reasoning [Allen&Koomen83, Allen91a, 91b]. We consider a plan as a partial ordering of actions; the actions having extent in time and space. We add spatial modifications to plan reasoning arguing that this is a dual of temporal reasoning (see for example [Cui etal92]).

1.1 Goals of the Research

First we develop a scheme for PEM based on spatial and temporal hierarchies to record the "where and when" of the domain. These hierarchies are developed for partitioning of space and time (s+t) and the inclusion of metric and point data. This research aims to develop mechanisms to implement the hierarchies for general reasoning about plans. Second, we integrate qualitative and quantitative information into the s&t hierarchical structures. The partitioning of s&t hierarchies will not be perfectly efficient (see Sect. 4) and this integration will augment the qualitative relations with quantitative information. This will enable representing and reasoning about disjunctive relations within the partitioning hierarchy. This representation is then used as the basis for developing an efficient computational approach to achieving approximate results for tasks such as PEM.

Exact results can be obtained for the intended dynamic and time constrained applications but this is not the expected mode of operation. The goal is a system which is able to respond in a timely and appropriate manner.

We use quantitative information on interval endpoints and durations to extend work in reasoning with hierarchical and abstract/expansion/aggregate data structures. The quantitative information is used within an approximate disjoint partitioning of the hierarchies (encoded with reference intervals) to preserve information. Quantitative point and duration values are assumed to be bounded intervals that encode the uncertainty of the extent of a domain object. An example using spatial-temporal information to reason about plans is presented. We also discuss this approach as applied to a Constraint Satisfaction Problem (CSP).

1.2 Background and Motivation

Recent work in plan generation addresses the reality that execution is no longer assumed to be in an idealized world where every action has its intended effect [Hanks90, A-I&S88]. The problem now becomes how to know what is happening or has happened and how to alter execution accordingly. Expectations of the world state are not always realized and exogenous events occur such that a sequential state transition model does not adequately reflect the complexity of the worlds in which agents will be expected to act. Implicit here is that events don't always occur when or where expected. There is some uncertainty as to the actual occurrence of an event or action. There is also uncertainty as to the spatial and temporal aspects of an event or action. We are concerned with this latter uncertainty.

Even with valid metric information, propagation through a constraint network and validation of the network may take more time than is available [Dechter et al89, Van Beek90]. Not only do we need to control the search, in both update and query, but in some applications an *anytime* response may be required [see, Dean&Boddy88]. Further, given more time a better response is possible.

One potential solution to the complexity of the problem is the use of abstraction hierarchies and reference intervals [Allen83, Koomen89] that in effect partition the search space of temporal and spatial information. The motivation here for considering space *and* time is that essentially everything, every action, every object and agent, must be somewhere at sometime. This is generally the minimum information we need to know and that is available during PEM. The combined knowledge of the two hierarchies further partitions the search space. The problem is to construct appropriate hierarchies. The use of user defined reference intervals and automatically generated reference intervals [Allen83, Koomen88, 89] has been proposed but problems remain with cross connections between reference intervals creating flattened structures. Planning techniques

such as abstraction [Kautz87, Tenenber88, Hanks92] may also provide structure within applications along with certain monotonic characteristics [Tenenber91] that make them attractive for PEM.

An approach to the reasoning needed to support anytime response is the relaxation of constraints in a constraint network [Mackworth77]. Constraints are given distance bounds between nodes in a graph [Dechter et al89]. This could also be applied to a spatial (3D) network although the result is not likely to be intuitive nor computationally attractive. Multiple constraints between nodes can be relaxed to a single aggregate constraint. This transforms the network into a simple constraint network with faster computational characteristics. The network might now allow values for which no singleton solution exists. We expand the constraints into convex intervals (*convexify* in [Ladkin86]) as "aggregate constraints". Each step of this relaxation includes all plausible values given as constraints. It may now include previously excluded values from when exact methods are employed. Plausible here means that it could be a member of a valid plan or network given some combination of constraints and relations. In [Shafer&Pearl90] *plausible reasoning* is offered as "reasoning that leads to uncertain conclusions because its methods are fallible or its premises are uncertain". Implicit is that there is evidence or reason to make the conclusion.

The differing requirements of PG and PEM require not only different representations of time and space but also the unification of the differing representations and information available. The problem is to combine the qualitative PG notions with the metric data of PEM. An abstraction hierarchy is capable of incorporating information at various levels of detail in both interval (qualitative) and point/metric (quantitative) forms. The use of space and time provides further structuring of information and at a level useful to the PEM problem in a domain independent way. This structuring then is able to support both uncertainty (as to interval bounds) and anytime (albeit approximate) response in support of PEM.

2 APPROACH TO THE PROBLEM

In a real world of time constrained situations, the larger driving question is: "what is the relation between a complex planning system and the real world". It is no longer realistic to consider a planner as only a static-world, omnipotent, plan generator [Allen&Schubert91]. We consider a planning system to consist of a deliberative or reasoning element, an execution element or agents and also a monitoring element. In short, *the planning system* needs to interact with the world. The approach we take to this interaction is to monitor the world in a flexible, adaptable way that can mesh real world observations with a projected world model. This is done in the basic parameters of the real world, i.e. space and time, and in

the constraints of executing in the real world, i.e. time-constrained.

Plan execution agents, or simply agents, that execute in the real world must be able to respond to unexpected external events in a timely and spatially coherent manner [Dennett87]. Temporal and spatial organization of knowledge can aid in monitoring and mediating an executing plan or an agent's decisions and recognition of significant events in constrained situations. The representation of temporal and spatial properties is integral to the way we are able to reason about such properties.

Specifically, we wish to look at the monitoring of an executing plan for which we have a projected world model for that plan. Reasoning about a plan's methods or goals is done by domain specific deliberative units and is not considered a part of the monitoring problem. We look at what information is true or known, when and where. The problem is that of (response) time and relevance of the new data. It is not practical to completely propagate each and every data point sensed or reported during monitoring against all known constraints. Also, data points might fit into one or more disjunctive or conditional projections (scenarios) of the executing plan. It might be more efficient to consider the union of these scenarios rather than each one separately. In short, not all data from monitoring should be assumed to be processed completely (excessive data bandwidth), nor would we wish to process all data to the same extent (data relevance).

The approach we take relies on hierarchical knowledge organization, aggregation of constraints, and integrating metric, point and absolute information into mostly qualitative and symbolic representations. We begin with a generated plan and a knowledge base of projected events, actions and facts based on the execution of that plan. This is assumed to consist of domain information that is temporally and spatially modified. The sentences in our data-base are triples of: *domain-object*, *space-object*, and *time-object*. The problem is now to determine when the projected course of events differs from, or can no longer be supported by, the observed state of the world. By combining qualitative and quantitative (metric and absolute) information we can monitor the progress of the plan execution vs the projection of the plan's intended actions and effects.

The temporal and spatial modifiers are organized into separate expansion hierarchies. The hierarchy is artificial in that it servers to cluster objects into reference intervals for time and space with well defined relations between the clusters. This network of qualitative constraints on the reference intervals, and the objects in the intervals, serve to partition the search space. Each object, including reference intervals, is then augmented with metric information as it becomes available. This information is used in combination with the qualitative network to enhance the overall expressiveness of the system. An aim

of this research is to include metric information from plan monitoring to enable the control of propagation and search in the network of s+t relations.

2.1 Monitoring and Updating

We choose to look at plan execution for several reasons. The most obvious is that generated plans are meant to be executed. Successors to the state-based models of [Fikes&Nilsson72, Green69] include abstract plan operations and partial plans that can only be fully completed or known upon execution [Wilkins88]. In other words, monitoring is necessary to know what or how to complete and execute in the plan. Learning systems such as in [desJardins92] seek general solutions for autonomous agents in complex domains. Our approach to monitoring is not as a stimulus to behaviors but as a rational component in the interaction between a reasoning system and the real world. This includes the need to consider the uncertain and dynamic character of the real world in which plans are executed.

2.2 Expansion Hierarchies

In both PG and PEM, abstraction hierarchies can be used to reduce the search space and number of possible plan instantiations or completions [Sacerdotti74, Tenenber88, Kautz87]. This method typically applies to actions, objects or types which have similar or common characteristics and then abstracts this commonality into a higher, generic or more encompassing classification or type. As an example we might abstract all the types of actions or series of actions that achieve some specific effect, call it X, into a pseudo-action (Abstract-achieve-X). This abstract pseudo-action is then just a symbol or place holder for one of the specific actions (specific-achieve-X1, specific-achieve-X2 etc) that actually achieve the effect X.

THE S&T HIERARCHIES

We propose a hierarchy structure that is slightly different from the abstraction presented above. The granularity of the parameters that modify a domain statement are variable within spatial and temporal hierarchies. In general, we allow for the increase or decrease of the range of domain statement modifiers. This affects the range over which these may be reasoned about or assumed.

In a hierarchy, an interval I is lower or subordinate to a reference interval I' if and only if:

1. I is a subset (included) in I' or
2. I is a subset of I'' and I' is a reference interval for I''.

One problem that can arise is that the hierarchy which is created is not structurally the one that is needed. The structure of the hierarchies should provide information. If there is excessive overlap or disjunction in the time or space constraints of domain statements, the purpose of the

hierarchy is somewhat defeated. The solution to this problem begins with the explicit addition of constraints and quantitative information on the bounds of the objects in the hierarchies and on the hierarchical structure itself. We focus on time and space as useful partitions of any KB since it is not often that we wish to know just of the existence of some object. We can then use this structure to reason about the KB in a variable and appropriate manner that may depend on the KB, the request or the specific data. It's a question of relevance.

THE AXIOMS OF S+T HIERARCHIES

We use a surface representation in the form of triples. Each form consists of a fluent, proposition or domain object D (the sentences of the domain) with spatial modifier S and temporal modifier T over which the domain statement is true. This is a syntactic variation of $\text{Holds}(D, S, T)$. The interpretation is that D holds at least somewhere over S and T . Negation of the fluent or object D is not allowed but only negation of a triple. This has the interpretation that D does not hold anywhere over (S, T) . Compared to Shoham's boundary condition $K(t, [\neg]p)$ an additional space parameter is added and negation moves outside [Shoham86]. The interpretation of negated triples (as $\neg p$) is the same but non-negated triples have the "plausible" semantics stated above.

$$\forall S', S, T, T' (S' \supseteq S, T' \supseteq T) . [D S T] \Rightarrow [D S' T'] \quad [A1]$$

$$\forall S', T' [D S' T'] \equiv \exists S, T (S' \supseteq S, T' \supseteq T) . [D S T] \quad [A2]$$

As a logical consequence of A2 the following theorem for negated triples is stated.

$$\neg [D S' T'] \Rightarrow \forall S, T (S' \supseteq S, T' \supseteq T) . \neg [D S T] \quad [T1]$$

The first axiom states that if a domain sentence is true somewhere in (S, T) then it is true in all expansions that include (S, T) . Conversely, if we know that a domain sentence holds over (S, T) then we know of the existence of some refinement of (S, T) where the sentence holds. Theorem T1 states the implication of a sentence that does not hold anywhere within (S, T) . Negated triples mean that the domain sentence is not true in all refinements of (S, T) . There is similar to standard Kripke possible world semantics. Also, *facts, events, properties* and other types of domain object types are not introduced. This presents triples as unitary notions of spatial and temporal propositions and follows [Shoham86a] in that regard.

3 S&T Examples

The following example shows the monitoring and reasoning about agents using a spatial expansion hierarchy and temporal information with metric data. System operation is introduced first, then a simple example showing the usage of temporal and spatial hierarchy information in execution monitoring. Next an execution with an indeterminate situation is presented along with how a simple spatial constraint is used to

resolve a potential conflict. This resolution provides a simpler, less complex constraint than temporal constraints alone would require.

3.1 System Operation

The execution monitoring system provides a mechanism for interpreting the status of an executing plan in terms of s&t consistency. It provides a means of integrating metric and point data and also vague (in terms of s&t parameters) information into various types of plans. This includes qualitative and quantitatively constrained, partially ordered and abstract plans.

Given an update (sensor report etc) of the form $[D s t]$, where s and t are points, is there a $[D S T]$ s.t. $s \in S$ and $t \in T$ and $[D S T]$ is plausible in both the real world and the projection of the executing plan. Also, given an update of the form $[D S' T']$ where D holds of at least some of S' and T' , is there a $[D S T]$ s.t. $S' \supseteq S$ and $T' \supseteq T$ and $[D S T]$ holds?

Updates relative to the s&t hierarchy are made for each triple that is consistent. The bounds on the start time and finish time of the appropriate interval T are modified as appropriate and also the bounds on the spatial modifier S . This is done in conjunction with what duration and scope information is known about S and T respectively.

For example (with t being an initial report of an event):

$$[T.start < t] \ \& \ [T.finish > t]$$

$$[t + dur.max > T.finish] \ \& \ [t - dur.max < T.start]$$

For T in T' (a reference interval), updates to T' reflect the constraints imposed by T , the relation between T and T' and metric information. Also any interval relations within T' are updated using an Allen style interval reasoner and the propagation of metric constraints occurs [Kautz&Ladkin91]. If there is no change to T' and T has no overlapping relations or metric constraints with intervals outside of T' then the update procedure is finished. If T' is updated (bounds change) or T has overlapping relations outside of T' , then the updating proceeds on T' similar to that of T but at the next level in the hierarchy. The update of metric relations outside of T' also needs to be checked for consistency.

3.2 A Transportation Domain

In this example, we start with a symbolic, abstract plan in a simple transportation domain. The domain consists of two cities connected by segments of train tracks which make-up various paths between the cities (see [3.2]). Traveling over these segments of track are two train engines that operate as SA agents. We assume the agents have a map of the domain. An agent's task in this domain will consist of traversing a path from one city to another. We will monitor the agent's actions and the spatial-temporal aspects of plan execution. This scenario is

complex enough yet general enough to demonstrate our approach while using a constrained space of the track segments that simplifies some details for now.

Consider the following plans for the agents to execute:

Agent *Eng1*'s task is to travel, via some route denoted *Path1*, from city A to City B. The time during which this plan is to be executed is called interval *T1*. A route here is a non-cyclic sequence of track segments. The primitive actions that we consider are at the level of segment traversal by the domain agents. Similarly, *Eng2* is to travel along a route *Path2* from city B to city A during a time interval *T2*. These agents are autonomous in path selection in that no constraints have been given for path selection. Likewise, the time intervals are initially unconstrained by the planner, although as we shall see the tasks and topography constrain the plans as execution proceeds.

The agents are making the routing decisions and may initially determine complete routes or postpone decisions until sometime during execution, perhaps relying on local conditions. *Eng1* begins with an abstract plan to reach the destination city over the unspecified route *Path1* during time *T1*. Let the plan or situation be represented to the plan monitoring component (PMC) as a triple: [Eng1 Path1 T1]. For the spatial-temporal monitoring component of the system this represents the fact that *Eng1* is at or in location *Path1* during time *T1*.

Since the agent must act on these abstract plans, concrete actions result. A representation of the plans specified to the agents is received by the PMC of the system. The PMC receives updates of the agents activities which consist of time and location information. The PMC must be able to monitor these activities and integrate this information with the overall plan. Further, we want to be able to determine if the information received by the PMC is consistent with its view of the world, i.e. its model, plans and projections. Additionally, unanticipated actions may be occurring and also need to be monitored for interaction with the executing plans. Only relevant information should then be directed to an agent. We are not considering agent-to-agent communication for now, nor other complex agent-agent interactions such as negotiations, recognition or acknowledgement as these are outside the scope of this work.

Consider now the high level abstract representation of the situation described above. In our projected knowledge-base there are the representations of *Eng1* and *Eng2*'s plan actions. The essential part for monitoring is the movement during the two time intervals and over the space *S* of the domain (see [3.2]). *S* represents all space in the domain and similarly there is a root time interval *T* that contains all other time intervals of interest. These are the most abstract or expansive values in their respective hierarchies.

Initial KB: [Eng1 Path1 T1] [Eng2 Path2 T2] [3.1]

T1 and *T2* are unconstrained time objects; essentially time intervals with additional metric information attached. The symbols *Path1* and *Path2* both denote the most general or least constrained spatial value: *S*. *S* is the expansion or generalization of all possible paths within the domain. We can represent any path (see [4.2] below) thru the domain as the disjunction of all segments making a path from CityA to CityB. In a more dynamic and complex domain it is reasonable to think that agents begin with abstract plans and leave the exact instantiations to be done later as necessary. The planning system later constrains the agent's paths based on local information or changing situations. Here we are calling spatial information "paths" when a more general terminology might be location or spatial scope of a domain object, predicate, event or action.

The AND/OR representation of all the paths is a relational constraint description of the spatially simple domain paths that connect CityA and CityB. This spatial domain is restricted to the segments and their connections. The explicit qualitative constraints between segments are limited to *equal*, *meets*, *met-by*, *contains* and *disjoint*. We see these are a subset of Allen's temporal interval relations [Allen83]. The relation *contains* is added for constraining the expansions in the spatial hierarchy. An example of this is: *S1/S2 contains S1*. This reinforces our intuition that temporal and spatial reasoning are similar and can be used and supported with similar techniques. More complex domains will make use of additional spatial relations (*above*, *with-in*, *below*, *north-of*, etc) as necessary. A spatial object in the domain is a segment or a sequence of path segments. This is shown below with implied ANDs while choice points are indicated explicitly by ORs.

$$[S \Rightarrow (OR (S1 S2) (S3 (OR (S4) (S5 S6)) S7))] \quad [3.2]$$

In the example given here we have a spatial expansion hierarchy for this domain that partitions the space *S*. Each expansion space is disjoint from each other at the same level. Each node within the space is labeled with the track segment or path abstraction represented by that node. Each non-leaf node is an abstraction or expansion of the nodes below it in the tree. The leaves of the tree indicate the possible exact instantiations of the space above it.

3.2.1 Example with no conflict

In this example, we start with the two abstract plan representations in [3.3] and the static spatial data of [3.2]. Each agent knows its own plans and the spatial information about the domain. The PMC of the planning system has this information plus receiving reports during execution time. From this situation the PMC needs to

recognize conflicts and the planning system needs to respond with additional constraints on the agents to ensure conflict free path selection.

At time 0900, agent Eng1 reports leaving city A and Eng2 reports leaves city B and the current track they are located on.

Added to KB: [Eng1 S1 0900] and [Eng2 S7 0900] [3.3]

These data require *Path1*, of [3.1], to have the value *S1/S2* at the most abstract level and *Path2* to point to the value *S3/S7*. The segments *S1* and *S7* are in completely disjoint subspaces of the *S* hierarchy. No spatial conflict is anticipated between *Eng1* and *Eng2* during the time intervals *T1* and *T2* and any further analysis will yield no additional information. If resources become constrained, the PMC could focus its monitoring resources to situations in which conflict is anticipated.

We see here the need to be able to incorporate metric information into the temporal representation. Our time objects are temporal intervals, symbolically represented but with additional quantitative information about end points and durations. Hence calling them objects rather than simply intervals. We have the absolute time values of [3.3] to incorporate into the initial interval representation. This is also true of the spatial representation as a numeric "mile post" could have been given instead of a track segment name. What is required is a method of determining the correct location in the hierarchy for the given metric information. Here *T1* and *T2* record a latest start time of 0900 for the intervals *T1* and *T2*. If durational information about the transit times of the segments is known then we could generate bounds on the end points of the time intervals. This would be useful in limiting unnecessary constraint checking. What should be noted here is that the reports coincide with the plan and that this argues for minimal action. It's what is expected. Any additional report that places *Eng1* on *S1/S2* and or *Eng2* on *S3/S7* does not force further checking of constraints.

3.2.2 Example with conflict

Starting with the same abstract plans and spatial information, let us consider another situation. The domain constraints precluded more than one agent to traverse a segment at any one time. Let the reports indicate that some conflict is possible but not inevitable. Each agent starts out on paths that may, when fully specified, be in some conflict.

[Eng1 S3 0900] and [Eng2 S7 0900] [3.4]

Given this as the initial KB, both *Path1* and *Path2* now point to the same node, namely *S3/S7*. There exists both conflict and non-conflict routes which might be taken given this initial report. The necessary information is

added based on the spatial decomposition of *S3/S7* and generate appropriate time intervals.

[Eng1 S3 T11] [Eng2 S7 T13] [3.5]
[Eng2 S7 T21] [Eng2 S3 T23]

Where *T11* and *T13* are contained during *T1* and likewise for *T21*, *T23* and *T2*

There is no simple, single temporal constraint that will ensure success, given what we know. By generating temporal intervals for each agent on both *S3* and *S7* and then constraining the appropriate intervals to be disjoint, we can avoid contention for *S3* and *S7*. These are the known segments that must be traversed. The problem is that the conflict, using temporal considerations, most likely occurs on segments *S4* or *S5* and *S6*. Using temporal constraints alone would over constrain the problem by generating disjoint time intervals for each agents actions while in *S4/S5/S6* space. This is unnecessary depending on the actual path taken in this region and we wish to avoid generating new and perhaps unnecessary constraints. Additionally the possibility of contention for *S3* or *S7* might be heightened by possibly delaying an agent from entering the *S4/S5/S6* region due to a disjoint constraint temporal constraint. Here, a simple spatial constraint is available and more desirable.

The simplest constraint is then a spatial constraint on each agent's path selection. Under the *S3/S7* node in the *S* hierarchy we constrain the agents selection at the first OR node choices. This way we make the minimum number of constraints necessary.

[Eng1 S4 T12] and [Eng2 S5/S6 T22] [3.6]

As noted before, *T11* and *T23* are disjoint intervals and the same for intervals *T21* and *T13*. If durational information about the transit times of the segments is known then we might not need to post the disjoint constraints and also do the subsequent consistency check. We would be able to determine that the intervals are disjoint based on statistical information about transit duration. Details are available in [Hoebel92].

We see from these examples that spatial and temporal constraints can be useful in the proper interpretation of reported data. This compound approach can also lead to simpler constraints than temporal reasoning alone. Monitoring leads to intervention (or not) at an appropriate time and in an appropriate manner.

4 Current and Proposed Work

Partitioning of the search space is one way to reduce the complexity of search and hence the time needed during query or update [Dean86]. In this section we employ techniques consistent with hierarchical knowledge

structure and show how metric information extends the restricted qualitative model.

4.1 Extending a Restricted Qualitative Model

In [Kahn&Gorry77, Allen83 and Koomen89 a reference hierarchy is proposed with the main motivation being to reduce the space requirements. Koomen differs in that he rejects the notion of a predefined reference hierarchy and also the notion that structure should be left to a higher level reasoning system. He describes a system where a "reasoning system itself could structure the network dynamically on the basis of posted and derived constraints, and do it without losing information"

One approach to an efficient implementation is to have a structure based on interval relations in the s+t hierarchies that are non-overlapping. Koomen refers to this as a "containment-based" reference hierarchy. We extend this interval based model in two ways. First developing procedures for overlapping relations to exist between intervals of non-containing (disjoint) reference intervals without flattening the hierarchies. Implicit in this is disjunctive constraints. Second we add metric, point and durational information which enables the management of the reference hierarchies while limiting propagation. This has attractive space requirements while providing computational mechanisms in moving beyond interval and relative duration information.

This extension now allows all binary (point and interval) relations to be represented while avoiding flattening the hierarchy. A full reasoner, interval, point, metric or combined, applied within reference intervals of a reasonable size is still very efficient [Koomen88, 89 Kautz&Ladkin91]. A full reasoner for spatial relations would be obviously more complex but it is reasonable to believe that a subset of spatial interval relations can be developed that is both adequately expressive and efficient [Cui etal92]. The problem is how to obtain and manage an appropriate s+t hierarchy. In the dynamic environment of PEM, hierarchies necessarily must be flexible to the changes and uncertainty that occur in the world. The semantics of the s+t expansion hierarchies provide the flexibility via the granularity of the s+t parameters. The separate handling of disjoint/contains and overlaps constraint relations is more fully adequate in terms of computation, expressiveness, the incorporation of metric point data and for exact or approximate reasoning than previously proposed systems.

4.2 Adding Metric Information: control of propagation and search

The inclusion of global constraints or chains of *overlapping* relations can lead to a rapid flattening of a reference hierarchy [Koomen89]. The problem generated by these overlapping interval relations can be overcome

by keeping the disjoint/contains reference hierarchy intact while explicitly and exactly encoding the overlapping relations which exist between domain intervals and reference intervals. This method allows a "meta-graph" of reference intervals for faster search while restricting propagation and consideration of the overlap relations only when appropriate. The reference hierarchy can be efficiently managed with the addition of metric and point information. Using this information, a disjoint/contains hierarchy can be created and managed even when such a structure does not exactly exist.

RELATIONS WITHIN A REFERENCE INTERVAL

Interval relations within a particular reference interval are intended to be reasoned about completely. A full reasoner for this task would use qualitative constraints as in a Allen style interval reasoner plus a point algebra and metric information. A system such as MATS appears to be able to handle this task in low order polynomial time and as such is considered adequate for the task. A table of qualitative interval relations is maintained while metric and point information is stored with each interval object. For intervals entirely within a single reference interval, all updates and queries concerning these intervals are assumed to be handled by the full reasoner. Each interval also maintains explicit overlaps information for overlapping relations outside the reference interval. This is for asserted information and does not imply that deduced relations are stored.

OVERLAPPING BETWEEN REFERENCE INTERVALS

Particularly during PEM, hierarchies need to be flexible and dynamic in structure. This is true even for those with an initial disjoint/contains structure. For example, updates to intervals within a reference interval may extend an interval into an overlapping relation. Consider the simple case of an action, event or goal of an executing plan having a longer duration than was initially conceived of in the plan. Updating the bounds of such an interval and its parent reference interval may now create overlapping conditions with other intervals. Rather than collapse the reference intervals of these now overlapping intervals into a single and larger reference interval, the overlapping relation is noted and reasoned about as an adjunct activity during search and propagation. Point and metric information can be used to determine the extent of the possible overlap. This information can prevent unnecessary computation by limiting search and propagation to the relevant areas of the hierarchy. Metric, point and crucially duration information allows the system to quantify the extent of overlap and respond accordingly.

UPDATING CONSTRAINTS, HANDLING ASSERTIONS (UNARY, BINARY, METRIC, QUALITATIVE)

New assertions or updates may require the interval or intervals concerned to be updated with new information. Any changes are first made to the interval objects

themselves. Changes then propagate downward as the updated interval may also serve as reference interval. A full reasoner is then applied to the entire reference interval containing the updated interval. After the full reasoner propagates constraints throughout the reference interval, affected intervals that serve as reference intervals propagate these changes downward. Finally, reference interval changes are propagated upward through the containing reference hierarchy and outward to sibling reference intervals. Note that procedures are called only when changes are made to intervals. No procedure is called on intervals outside the reference interval unless a change occurs that affects the interval. Also note that any assertion or propagation that effects an overlapping inter-reference interval relation is handled as a separate subprocedure during the propagation procedure that effects the overlapping intervals.

SUMMARY OF ASSERTION UPDATING STEPS

1. Update intervals (endpoint constraints, durations and explicit relations)
2. Propagate interval update downwards (recursively)
3. Apply full reasoner to all members of the intervals reference interval (those that change are queued)
 - 3.1 Update to overlapping reference intervals outside the initial update interval's reference interval. (keeping a queue of intervals that are changed)
4. Propagate changes to reference interval (and queue reference interval if changed)
5. Select next interval in queue.

In the case where we know that reference intervals may overlap but not know the extent of the overlap, we are faced with a choice. One method is a full reasoner used over the combined reference intervals to detect all possible relations. Although this is possible, it is not advisable in general since this is the exact problem of hierarchy flattening that we are trying to avoid. The simplest course is to do nothing and wait until information is obtained that reveals the extent of the overlap. A more balanced approach is to require some estimate of the duration of each domain object, activity, fluent, etc. This of course is only a middle ground between knowing nothing of the extent and of knowing exactly the extent. As the system reasons over longer spans in the hierarchy, the more the durational uncertainty is going to accumulate and tend towards ignorance of actual overlap extent.¹ In practice we expect the structure of the problem to be such that propagation is generally limited to local areas of the hierarchies. Overlapping relations will not tend to propagate endlessly. The CSP example in the next section shows how absolute times, even when dealing with expanded constraints, limit propagation quickly and provide accurate bounds.

¹ A separate issue to address is the characterization of this durational uncertainty.

USING METRIC CONSTRAINTS

When metric information, point information and duration information is added to symbolic interval constraints, the ability to determine exact relationships is increased. Consider for example three intervals A, B and C and the constraints *A overlaps B* and *B overlaps C*. A and C are related by any of the relations *before*, *meets*, or *overlaps*. We can determine if A does overlap C with the addition of metric and endpoint information for each interval. This is something that is not possible with strictly qualitative reasoning over intervals. Given bounds on the start, finish and duration times of the intervals, the determination of possible relations can be made. These bounds may or may not be the same as a "stated constraint" on the endpoints or the duration of an interval. We may be able to infer a tighter constraint. This is shown in the example in sect. 4.3.1. For this reason it may be desirable to track both stated bounds and an inferred (non monotonic) bounds.

Lets continue the case of intervals A, B and C as described above. With additional metric information about the start and finish times we may be able to determine a more exact ordering of the endpoints. We now impose the following three metric point constraints:

- [A.finish - B.start] < 30
- [B.finish - C.start] < 30
- [B.finish - B.start] > 60

The third constraint requires a minimum duration for interval B that requires A before C. This type of information is useful in the determination and management of of intervals in reference hierarchies. Consider now the case in which slightly different constraints allow for a finite amount of possible overlap. Consider the situation, in the notation of Allen, (A :o :m :b B :o :m :b C):

- [A.finish - B.start] < 35
- [B.finish - C.start] < 35
- [B.finish - B.start] > 60

These constraints now allow for A and C to overlap but only by up to 10 units. This does not require that A and C be contained within a single reference interval. We can place A and C in disjoint reference intervals, note the overlap and then use the explicit information when required. Here we can note the constraint between B and both A.finish and C.start. This information would propagate to the reference interval containing A and C. Subsequent updates to the reference intervals need not propagate across this "link" unless the "link" itself (the endpoints) is in some manner involved (changed).

Interval B, which might overlap both A and C, might be placed in either reference interval depending on some domain specific criteria such as causal connection or number and type of relations to the intervals in the containing reference intervals. In this way we can maintain the expansion disjoint/contains hierarchies but still deal efficiently with overlapping relations.

Additionally we can structure the hierarchies on domain information when such information is known and appropriate. This would appear to be an improvement over a predefined reference hierarchy.

4.3 A CSP model

In [Dechter et al89] an instance of the Constraint Satisfaction Problem is presented as a temporal reasoning problem. They present an approach that incorporates qualitative, metric and absolute time in a network of temporal constraints. The absolute times are stored as special nodes, nodes are temporal variables in general, and the constraints are added as arcs between variable nodes. In a longer version of this paper a transformation of the CSP is formulated that is consistent with the notions developed in the s+t hierarchies presented in this paper. First, a network is given for the example as presented in [Dechter et al89] and then a network where absolute time/arc constraints are reformulated as constraints (bounds) on node values. The special absolute time nodes are eliminated and the absolute times internalized at the variable nodes. This is done to avoid making a fully connected graph of arc constraints. Propagation of absolute times can be interrupted and restarted as they serve as something of a local constraint caching mechanism. The metric information itself can be used to control propagation and to determine currently plausible values for a node. An small extended example shows propagation of values to be localized and efficient within the hierarchy. The final model may return a weaker constraint interval than if exact methods are used. The absolute times are used to bound this weakness. An example is shown that transforms multiple arc constraints in a Temporal CSP (TCSP) into single arc constraints representative of the Simple TCSP. The results of this relaxed form of TCSP for approximate reasoning si compared to exact singleton solutions.

We have presented the outline of a representation that allows for approximate reasoning and can clearly incorporate symbolic, metric and approximate or uncertain information. We do this in a uniform way with both space and time and thus allow for easy application to scheduling and routing problems, which is to say that it applies to plans that have all 4 dimensions as part of their solution.

4.4 Summary

Limiting the expressiveness of the qualitative model limits the search required for queries and updates of the (basically) qualitative model. We start with this model and defined two types of intervals and give the constraint relations allowed between them. An extended model is presented that seeks to overcome the limited expressiveness of the first model. It uses absolute and metric time, durations, and endpoint relations to achieve control of propagation and capture the full expressiveness of relations while retaining the graph search information

of the restricted disjoint/contains model and the efficiency of metric comparisons.

Finally we discussed the general techniques of an expansion hierarchy as an approximate solution to a metric constraint satisfaction problem. The goal is to produce a more fully expressive spatial-temporal model that is efficient in general and has anytime characteristics. It appears that the efficiency needed for real-time response is in conflict with the complexity of a more expressive temporal reasoner [Vilain&Kautz86, others] but can be overcome by exploiting structure and with approximate methods.

I gratefully acknowledge the Rochester Temporal Reasoning group and Leo Hartman for what sanity is presented here. All errors are clearly my own.

References:

[Allen 83] James F. Allen *Maintaining Knowledge about Temporal Intervals*. Communications of the ACM26 (no. 11): 832-843

[Allen&Koomen 83] Allen J.F. and J. A. Koomen *Planning Using a Temporal World Model*" Proceedings of the 8th IJCAI pps 741-747. Karlsruhe, W.Germany August 1983

[Allen 91a] James F Allen, *Planning as Temporal Reasoning*, 2nd Conference on Knowledge Representation [KR91] April 1991, Boston Mass.

[Allen 91b] Allen, James; Kautz, Henry; Pelavin, Rich and Josh Tenenber "Reasoning about Plans" Morgan Kaufman Publishers 1991

[Allen&Schubert 91] Allen, James. and Len Schubert; "The Trains Project", Univ. of Rochester, Dept. of Comp. Sci., TRAINS Technical Note 91-1 May 1991

[Ambrose-Ingersal&Steele88] in "Readings in Planning" Eds. Allen, Tate and Hendler; Morgan Kaufman 1991

[Cui etal92] Cui, Z; Cohn, A & D. Randell "Qualitative Simulation Based on a Logical Formalism of Space and Time" Proceedings AAI-92, San Jose CA. 1992

[Dean86] Thomas Dean, "Intractability and Time Dependent Planning" The1986 Workshop on Reasoning about Actions and Plans; Morgan Kaufman Publishers.

[Dean&Boddy 88] Thomas Dean and Mark Boddy "*An Analysis of Time-Dependent Planning*" Proceedings of the 7th AAI, Aug. 1988 ppg 49-54

[Dean&McDermott87] Dean T.L. and McDermott D.V. *Temporal Date Base Management AI Journal* 1-55 1987

- [Dechter et al89] Dechter, R., Meiri, I., and Pearl J. *Temporal Constraint Networks* Proceedings of the Conference on Knowledge Representation [KR89] pps 83-93 Toronto, Canada May 1989
- [desJardins92] Marie desJardins, "PAGODA: A Model for Autonomous Learning in Probabilistic Domains" Univ. of Calif at Berkeley, UCB/CSD 92/678 1992
- [Dennett87] D. Dennett "Cognitive Wheels: The Frame Problem in AI" in *The Robot's Dilemma* Ed. by Z. Pylshyn Ablex Publishing, Norwood NJ 1987
- [Green69] Green, Cordell "Application of Theorem Proving to Problem Solving" Readings in Planning, Eds. J. Allen, J. Hendler and A. Tate pgs 67-87 Morgan Kaufman Publishers 1991.
- [Hanks 90] Steve Hanks "Practical Temporal Projection" Proceedings of the 8th AAI, Boston MA August 1990.
- [Hoebel92] Louis Hoebel, Tech. Report (forthcoming)
- [Kahn&Gorry77] Kahn, K. and G. Gorry "Mechanizing Temporal Knowledge" AI Journal 87-109, 1977
- [Kautz 87] Henry A. Kautz "A Formal Theory of Plan Recognition" Tech. Rep. TR-215 University of Rochester, Dept. of Computer Science, May 1987
- [Koomen88] Koomen J.A. *The TIMELOGIC Temporal Reasoning System*, Tech. Rep TR-231 University of Rochester, Dept. of Computer Science, October 1988
- [Koomen88a] Koomen J.A. "Reasoning about Recurrence" Phd Thesis University of Rochester, Dept. of Computer Science, October 1988
- [Koomen89] Koomen J.A. "Localizing Temporal Constraint Propagation" Proceedings of the 1st Conference on Knowledge Representation Toronto Canada 1989 [KR89] 198-202
- [Ladkin86] Peter Ladkin "Primitives and Units for Time Specification" Proceedings of the 5th AAI, pps 354-359, Philadelphia, PA. August 1986
- [Mackworth 77] Allan K Mackworth "Consistency in Networks of Relations" AI Journal pps 99 -108, 1977,
- [Malik & Binford83] Malik and Binford *Reasoning about space and time*. [IJCAI 83]
- [McDermott82] Drew McDermott, *A Temporal Logic for Reasoning about Actions and Plans* Cognitive Science 6(2), 1982 pps 71-109.
- [Sacerdoti74] Earl Sacerdoti, "Planning in a Hierarchy of Abstraction Spaces" Artificial Intelligence 115-135, 1974
- [Shafer76] Glen Shafer, "A Mathematical Theory of Evidence" Princeton Press, 1976
- [Shoham86] Yoav Shoham, "Chronological Ignorance" Proceedings of the 5th AAI-86 pps 389-393, Philadelphia, PA. August 1986
- [Shoham86a] Yoav Shoham, "Temporal Logics in AI" Artificial Intelligence Vol. 33, No. 1 September 1987
- [Tenenber88] Josh Tenenber "Abstraction in Planning" Phd Thesis, University of Rochester Dept. of Computer Science TR-250; May 1988
- [Teneber91] [Knoblock, C; Qiang, Y. and J.Tenenber "Characterizing Abstraction Hierarchies for Planning" AAI-91. pps 692-697 Anahiem CA, July 1991
- [Traum and Allen 91] Traum, David R. and James F. Allen, "Causative Forces in Multiagent Planning" in *Decentralized AI 2* Editors Y. Demazeau and J. Muller, Science Publishers B. V.
- [Vilain&Kautz86] Vilain, M. and H. Kautz, "Constraint Propagation Algorithms for Temporal Reasoning" AAI-86 pps 377-382, Philadelphia, PA. August 1986
- [Schubert&Miller90] Schubert L. and Miller S., *Time Revisited* Computational Intelligence 1990
- [Van Beek 90] Peter Van Beek, Reasoning about qualitative temporal information. Proceedings of the 8th AAI; Boston, Mass. August 1990.
- [Wilkins88] David E. Wilkins "Practical Planning" Morgan Kaufman Publishers, San Mateo Calif. 1988